

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Detekce dopravních značek z obrazu kamer  
zabudovaných ve vozidlech**  
(trojúhelníkové značky)

**Traffic Signs Detection from the Cameras  
Built in Vehicles**  
(Triangular Signs)

## Zadání diplomové práce

Student:

**Bc. Jiří Pánek**

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Detekce dopravních značek z obrazů kamer zabudovaných ve vozidlech  
(trojúhelníkové značky)  
Traffic Signs Detection from the Cameras Built in Vehicles (Triangular  
Signs)

Zásady pro vypracování:

Detekce dopravních značek z jedoucího vozidla je významnou úlohou. Výsledky detekce mohou být využity nejen ve prospěch řidiče vozidla, v němž byla detekce provedena, ale mohou také sloužit, prostřednictvím informačního systému, pro kontrolu stavu a aktuálnosti značení. Cílem diplomové práce je vyvinout metodu pro detekci dopravních značek trojúhelníkového tvaru.

V diplomové práci proveďte:

1. Seznamte se se současným stavem v dané oblasti a shrňte jej v rešeršní části práce.
  2. Na základě získaných poznatků navrhnete metodu pro detekci dopravních značek trojúhelníkového tvaru.
  3. Navrženou metodu naimplementujte a řádně experimentálně proveďte. Výsledky shrňte v textové části práce.
- Implementaci proveďte v jazyce C++; můžete rovněž využít knihovny OpenCV.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

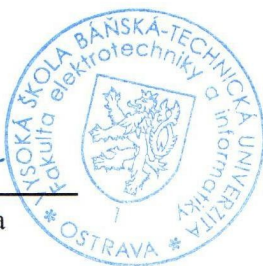
Vedoucí diplomové práce: **doc. Dr.Ing. Eduard Sojka**

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



doc. Dr.Ing. Eduard Sojka  
vedoucí katedry



prof. Ing. Ivo Vondrák, CSc.  
děkan fakulty

## **Prohlášení:**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.  
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě, dne 7.5.2010

Podpis: .....

Rád bych na tomto místě podekoval především vedoucímu práce za konstruktivní kritiku a připomínky při psaní této práce. Slova díků patří také mým rodičům a přátelům za morální podporu ve chvílích nejtěžších.

## **Abstrakt:**

Cílem práce bylo navrhnout a implementovat algoritmus pro detekci trojúhelníkových dopravních značek v obraze. Takovéto algoritmy mohou být použity nejen ke kontrole stavu dopravního značení, ale v podobě systémů podpory řidiče mohou zvyšovat bezpečnost provozu v dopravě. Pro nalezení polohy značky v obraze je využito barevné segmentace založené na prahování jednotlivých barevných kanálů. Nalezené dopravní značky jsou klasifikovány pomocí umělé neuronové sítě. Použití těchto dvou metod by mělo přinést požadované výsledky při zachování rychlosti, která umožní nasazení aplikace v reálném čase. Implementovaná aplikace umožňuje nastavení potřebných parametrů algoritmu a uživateli poskytuje informace nejen o výsledcích analýzy obrazu, ale i o jejím průběhu.

## **Klíčová slova:**

segmentace, prahování, umělá neuronová síť, dopravní značení, algoritmus, detekce, klasifikace, konvoluce, barevný prostor

## **Abstract:**

The aim of this work was to propose and implement an algorithm for detection of triangular traffic signs in an image. This kind of algorithm can be used not only for traffic signs condition control, but also as a driver assistant system improving traffic safety. Color segmentation based on each color channel thresholding is used for localization of the sign in an image. Artificial neural network classifies localized traffic signs. Using two methods mentioned above should ensure the speed empowering real-time using. Implemented application allows setting up necessary parameters and provides the results of image analysis as well as analysis progress information to user.

## **Keywords:**

segmentation, thresholding, artificial neural network, traffic signs, algorithm, detection, classification, convolution, color space

# Seznam použitých zkratk a symbolů

ABS	Anti-lock Braking System
ASR	Anti-Slip Regulation
EEG	Elektroencefalogram
EKG	Elektrokardiogram
ESP	Electronic Stability Program
FANN	Fast artificial Neural Network
HSV	Hue Saturation Value
NN(NS)	Neural Network (Neuronová Síť)
RGB	Red Green Blue
UNS	Umělá neuronová síť

## Obsah:

1 Úvod.....	1
1.1 Členění práce.....	2
2 Existující řešení.....	4
2.1 Automatická klasifikace dopravních značek.....	4
2.2 Blaupunkt TravelPilot.....	4
2.3 Automatické rozpoznávání dopravních značek v obraze.....	5
2.4 Rozpoznávání silničního značení z jedoucího vozidla.....	5
2.5 Rozpoznávání dopravních značek v reálném čase.....	6
3 Dopravní značení.....	8
3.1 Trojúhelníkové dopravní značky.....	10
3.2 Rozdíly v dopravním značení.....	10
4 Návrh systému.....	13
4.1 Detekce značek.....	13
4.1.1 Filtrace.....	13
4.1.2 Převod barevného prostoru.....	14
4.1.3 Segmentace.....	14
4.1.4 Detekce trojúhelníků.....	14
4.2 Klasifikace značek.....	15
5 Detektor - použité algoritmy.....	16
5.1 Převod do HSV prostoru.....	16
5.2 Filtrace.....	17
5.2.1 Konvoluce.....	17
5.2.2 Morfologická filtrace.....	20
5.3 Prahování.....	21
5.4 Segmentace.....	21
5.5 Hledání kontur.....	21
5.6 Freemanův kód.....	22
5.7 Prokládání polynomem.....	23
5.8 Warping.....	23
6 Klasifikátor – neuronová síť.....	25
6.1 Umělá neuronová síť.....	25
6.1.1 Neuron.....	26
6.1.2 Vícevrstvá neuronová síť.....	27
6.1.3 Metoda učení - Backpropagation.....	28
6.2 Trénovací množina.....	28
6.3 Trénování klasifikátoru.....	29
7 Testování výsledné aplikace.....	31
7.1 Testovací data.....	31
7.2 Intervaly hodnot pro segmentaci.....	31
7.3 Výběr filtrů.....	33
7.4 Úspěšnost detekce značek.....	35

7.5 Úspěšnost klasifikace značek.....	35
7.6 Rychlost aplikace.....	37
8 Ovládání aplikace.....	38
9 Závěr.....	40
Seznam ilustrací:.....	42
Příloha: CD.....	45

# 1 Úvod

S rozvojem vědy a techniky neustále přibývá zařízení a technologií, které ve svém životě člověk využívá. Většina těchto novinek je určena ke zjednodušení a usnadnění života nebo k zajištění bezpečnosti. Výjimkou není automobilový průmysl, ba právě naopak. V tomto odvětví se s různými vynálezy a technickými novinkami setkáváme velice často a mnohdy běžný uživatel vozu ani neví, k čemu všechna ta zařízení a přístroje vlastně slouží nebo jak přesně fungují. S rozvojem výpočetních technologií a především pak s jejich masovým rozšířením do všech možných oblastí lidské činnosti dochází k jejich zlevňování. Díky tomu je dnes v každém novém vozidle integrován alespoň jeden systém, který je řízen mikroprocesorem.

Díky neustále rostoucímu výpočetnímu výkonu jsme se pomalu dostali do fáze, kdy je možné do automobilu integrovat výpočetní jednotku, která dokáže, a to i v reálném čase, analyzovat a vyhodnocovat data z mnoha snímačů. Mezi tyto snímače patří zařízení, jako jsou ultrazvuková či mikrovlnná čidla. Čím dál častěji se však setkáváme s tím, že výrobce do svých vozů zabudovává i kamery. Některé jednodušší systémy jsou určeny pouze k tomu, aby řidič viděl tam, kde to jinak není možné – především za vozidlo. Jiné firmy vyvíjejí systémy, které snímají prostor před vozidlem infra kamerou a uživateli tak umožňují lepší vidění za horších optických podmínek. Sofistikované systémy pak umožňují analýzu nasnímaných dat, která umožňují například detekci překážky před vozidlem nebo se ve snímcích hledá dopravní značení. Všechny tyto různorodé systémy mají společný účel, kterým je především zvýšení bezpečnosti silničního provozu a minimalizace škod při kolizi. Často bývají označovány jako Driver Support System.

Analýza obrazu snímaného kamerou v jedoucím autě je jistě krok správným směrem, přinášející mnohé výhody a mající mnohá další využití. Řízení auta je totiž proces, který je závislý především na tom, jak řidič vnímá vizuální podmínky z okolí. První pokusy o vytvoření takového systému se objevily v roce 1986 v podobě projektu PROMETHEUS (PROgramM for a European Traffic with Highest Efficiently and Unprecedented Safety), na kterém se podílela téměř dvacítko evropských zemí, ve spolupráci s několika výrobci osobních automobilů.

Ve spojených státech započaly výzkumy v rámci NAHSC (National Automated Highway System Consortium) koncem roku 1994 a o další vývoj pak usilovala iniciativa IVI(Intelligent Vehicle Initiative), založená v roce 1997.



V Japonsku se první pokusy o podobné systémy vyvíjí již od roku 1987, od roku 1996 například v podobě projektu AHSRA (Advanced Cruise-Assist Highway systém Research Association).

Pokud systém zaznamená výstražnou dopravní značku, může pomocí informace na čelním panelu upozornit řidiče. V případě zaznamenání značky, která označuje nebezpečný úsek, může být automaticky zapnut některý z bezpečnostních systémů, jako je ABS, ASR, nebo ESP. V případě značení omezujícího maximální povolenou rychlost, může systém omezit rychlost vozu a tím řidiče uchránit před případnou pokutou.

Kromě prospěchu řidiči vozidla, může být detekce dopravního značení vhodná i pro třetí stranu. Například tvůrci mapových podkladů mohou začlenit dopravní značení do map, což může vést ke zlepšení výpočtů nejkratších či nejrychlejších tras. V takovýchto mapách je pak možné vyčíst, která ulice je jednosměrná, nebo kde je jaké dopravní omezení. Podniky starající se o dopravní značení se mohou dozvědět, kde je nějaká dopravní značka poškozena a jejím opravením mohou opět přispět ke zvýšení bezpečnosti. Oba tyto případy však vyžadují možnost přenosu informací mezi vozidly a zpracovateli těchto dat. Sice to ještě není úplně běžné, ale již teď některá vozidla mají připojení k internetu či podobné bezdrátové síti. Budou-li v budoucnu všechna vozidla připojena sítí se zpracovatelem dat, bude možné výsledky analýzy ihned odesílat.

## ***1.1 Členění práce***

Kapitola 2 stručně popisuje několik, již existujících, řešení. Četnost užití algoritmů pro detekci a klasifikaci v nalezených řešeních shrnují grafy v závěru kapitoly.

Třetí kapitola se věnuje dopravnímu značení. Nejprve se věnuje historii a postupnému vývoji, následuje popis trojúhelníkových dopravních značek a na závěr jsou zmíněny rozdíly mezi naším značením a dopravním značením sousedních států. Na úplném konci kapitoly je vyobrazen kompletní seznam rozpoznávaných dopravních značek.

Stručným popisem navrhovaného algoritmu se zabývá kapitola 4. Fungování celého systému a jeho dílčích částí je popsáno slovně a zachyceno na diagramech.

Kapitola pátá se věnuje přesnějšímu popisu algoritmů a technik použitých pro detekci polohy značek v obraze.

Úvod 6. kapitoly je věnován popisu struktury a fungování neuronové sítě. Jednotlivé podkapitoly se pak zaměřují na popis vytvoření trénovacích dat pro neuronovou síť, jejímu učení.

Sedmá kapitola prezentuje dosažené výsledky implementovaného systému po předložení testovacích dat.

Závěrečná kapitola shrnuje dosažené výsledky a navrhuje možná vylepšení a rozšíření popsaného algoritmu. Na konci kapitoly je pojednáno o možném vývoji a využití podobných systémů v budoucnu.

## 2 Existující řešení

Je možné nalézt poměrně hodně článků a prací, které se daným tématem zabývají. Drtivá většina se však zaměřuje pouze na část rozsáhlého problému, kterým detekce dopravního značení bezesporu je. Podívejme se na pár vybraných řešení, z nichž některá si již našla cestu ke komerčnímu využití.

### 2.1 Automatická klasifikace dopravních značek

Diplomová práce studenta Pavla Vaclíka z ČVUT v Praze, která byla oceněna Cenou firmy Siemens. K rozpoznání umístění dopravní značky ve snímku využívá informaci o lokálních orientacích hran[1]. Funkce systému je rozdělena do dvou kroků. Při detekci dochází ke hledání geometrických tvarů, které mohou představovat dopravní značku.

Druhým krokem je určení typu dopravní značky, kterému nalezená oblast odpovídá, pomocí statistického klasifikátoru. Metoda pro detekci má výhodu v tom, že nevyužívá barevné informace, což je výhodné především za horších optických podmínek. Při klasifikaci je využito příznaků, což jsou charakteristiky, které se pro různé značky liší. Ve fázi učení je použito na 2000 obrázků více než 40 typů dopravních značek. Klasifikátor lze označit za rozhodovací strom. Typ značky je postupně upřesňován, což znamená, že nejprve je značka zařazena do skupiny a teprve když je piktogram čitelný, je určen i typ.

Nejvýznamnějšími částmi práce je nejen typ klasifikátoru (neparametrický jádrový klasifikátor), ale především učící algoritmus, který je rychlejší než algoritmy užívané u neuronových sítí. Výsledná aplikace je schopna nalézt značky během jednotek sekund a určit typ během desítek milisekund. V závislosti na kvalitě vstupních snímků, je dosahováno 95% spolehlivosti klasifikace.

### 2.2 Blaupunkt TravelPilot

TravelPilot je název navigace, která se na první pohled příliš neliší od navigací ostatních výrobců. Při bližším prozkoumání však zjistíme, že obsahuje vestavěnou kameru, která snímá scénu před vozidlem. Obraz z kamery je promítán na display a směr jízdy je vyznačován pomocí šipek v příslušných jízdních pružích. Druhou funkcí je detekce dopravních značek omezujících maximální povolenou rychlost. Bohužel výrobce neuvádí, jaké algoritmy pro zpracování obrazu využívá. Z recenzí se dočteme, že někdy nedojde k detekování dopravní značky[2]. K tomu dochází především za horších světelných podmínek jako je tma nebo protisvětlo.

## 2.3 Automatické rozpoznávání dopravních značek v obraze

Autor, John Hatzidimos, v této práci využil ke hledání značek v obraze metodu barevné segmentace, která je následována hledáním hran[3]. Dále se v obraze vypočítávají souřadnice umístění dopravních značek a hledají linie, k čemuž je využita Houghova transformace. Nalezené značky se podrobují kontrole tvaru, což například pro trojúhelníkové dopravní značky znamená měření úhlů svíraných jejich stranami. Aby byla značka dále zpracována, musí naměřené úhly pocházet z rozmezí od 50°, do 70°.

Pro snazší rozpoznání je pak každá nalezená značka upravena pomocí affíní transformace, čímž se kompenzuje její natočení vůči kameře při snímání. Pro porovnávání značky se vzory se využívá korelačního koeficientu a pracuje se nezávisle s jednotlivými barevnými složkami obrazu. Jak sám autor připouští, bylo by zřejmě lepší využít v některých algoritmech obraz převedený do HSI, namísto klasického RGB barevného modelu.

## 2.4 Rozpoznávání silničního značení z jedoucího vozidla

Zde bylo k nalezení značky v obraze opět použito barevné segmentace a to v HSI barevném prostoru[4]. Následně je vzniklý obraz upraven pomocí erozivního a dilatačního filtru, což má za následek redukci šumu. Zjištěním velikosti souvislých oblastí je algoritmus schopen vyloučit z dalšího zpracování ty oblasti, které s největší pravděpodobností představují zbylý šum, nebo příliš vzdálené dopravní značky. V poslední fázi vyhledávání značky je provedena detekce hran.

Pro zjištění konkrétního typu dopravní značky autor využil klasifikaci na základě porovnávání se vzory. V tomto algoritmu je použit vztah pro výpočet Yuleova koeficientu:

$$Yule = \frac{n_{11}n_{00} - n_{10}n_{01}}{n_{11}n_{00} + n_{10}n_{01}} \quad (1)$$

kde  $n_{ab}$  představuje počet pixelů, které nabývají hodnotu  $a$  ve vzoru a současně hodnotu  $b$  v porovnávaném obraze.

Autor uvádí celkovou dobu pro zanalyzování jednoho snímku okolo 200ms. V případě, kdy je v obraze mnoho oblastí s červenou barvou, se čas zpracování stává až dvojnásobným.

## 2.5 Rozpoznávání dopravních značek v reálném čase

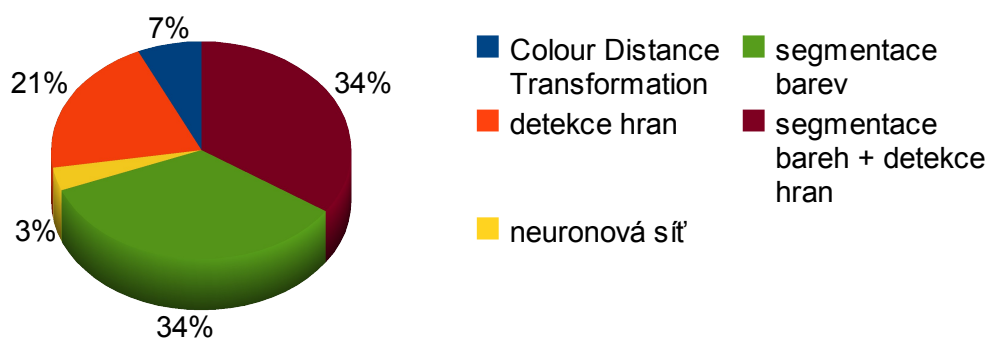
M. Taha Khan se v této práci nezaměřil na hledání pouze jedné barvy při segmentaci obrazu, ale rovnou vyhledává všechny barvy, které se ve švédském značení používají[5]. Jedná se o červenou, žlutou, modrou, bílou a černou barvu, jejichž intervaly v HSV barevném prostoru zachycuje následující tabulka. Intervaly jednotlivých barevných složek jsou normalizovány na rozsah hodnot od 0 do 255.

barva	H – odstín	S – sytost	V – jas
červená	0-10,210-255	127-255	127-255
žlutá	20-45	110-255	90-255
modrá	120-180	127-255	127-255
bílá	0-255	166-255	90-255
černá	0-255	0-230	0-63

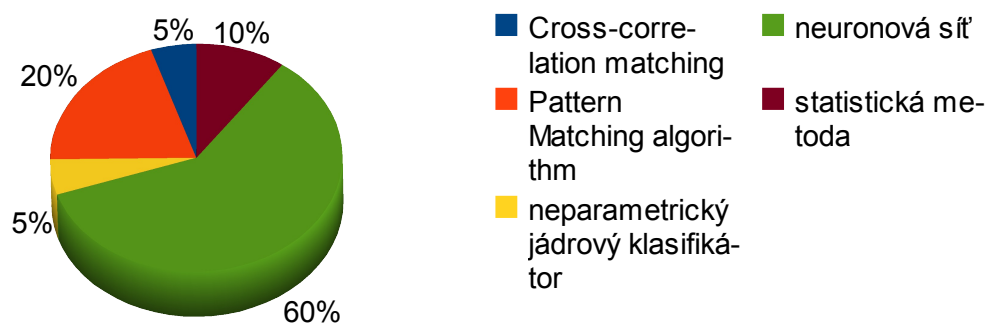
Tabulka 1: Intervaly barevných složek

Po nalezení segmentů odpovídajících jednotlivým barvám, je použit filtr split-and-merge, který odstraní šum. Dále se v obraze hledají komponenty, což jsou větší shluky pixelů požadované barvy. Příliš malé nebo naopak příliš velké komponenty se dále nezpracovávají. Dále už následuje identifikace typu značky na základě rozpoznávání vzorů (pattern matching). Autor uvádí, že jeho systém je schopen značku správně rozeznat s pravděpodobností 95% a to v průměrném čase 0,15s.

Jak je patrné z jejich popisu, nalezená řešení lze dělit do skupin podle využití metody segmentace a podle metody klasifikace. Tato rozdělení zachycují následující grafy. Uvedme ještě, že celkový počet nalezených a v grafech znázorněných řešení byl 29.



Graf metod pro vyhledávání



*Graf metod pro klasifikaci*

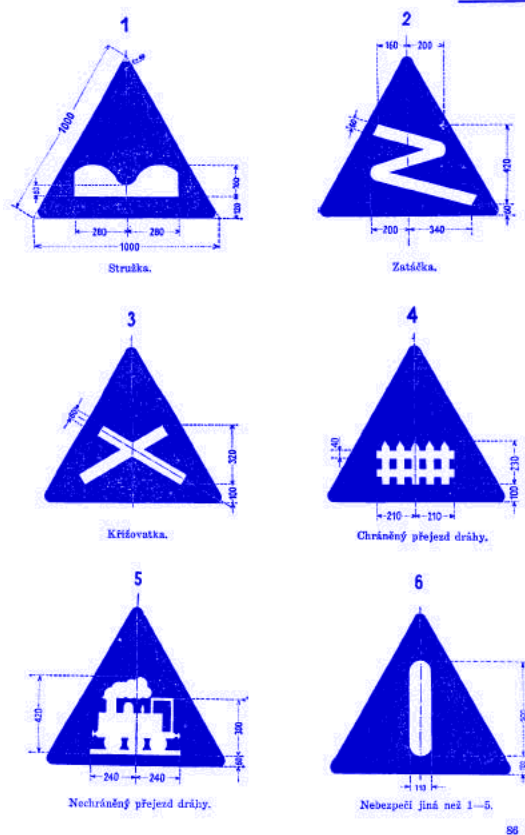
### 3 Dopravní značení

Prvopočátky užívání značení sahají hluboko do historie. Již 120 let př. n. l. se začalo užívat milníků, které udávaly vzdálenost od Říma [6]. V průběhu středověku se pokračovalo v tomto duchu, tzn. značení vzdáleností a směrů. Ruský car Petr Veliký počátkem 18. století nechal zavést značky, které udávaly vzdálenost mezi jednotlivými městy a byly označovány jako tzv. versníky. Některá užívaná značení neuváděla vzdálenost v délkových, ale časových jednotkách.

S příchodem vynálezu automobilu a s jeho postupným rozšířením se začala projevovat potřeba nového typu značení. Od počátku 20. století se postupně objevovaly snahy o zavedení nejen lokálních, ale i celoevropských systémů značení. Přes veškerou snahu se však dopravní značky v jednotlivých státech více či méně odlišují.

U nás se zavedly první dopravní značky v roce 1935 a šlo o šest typů značek výstražných (obr.3.1). Další značky přibýly spolu se zavedením pravostranného silničního provozu roku 1938. Nahrazení modré barvy za zřetelnější červenou v kombinaci s černými piktogramy přišlo s vyhláškou ministerstva vnitra ČSSR č. 141/1960 Sb. ze dne 3. září 1960 (obr.3.2). Dále se již existující značky příliš neměnily, ale zato přibýlo několik nových.

Postupně se tak značky vyvinuly až do dnešní podoby, která je přesně určena příslušnými vzorovými listy [7]. Navíc byl značkám přidán bílý lem a vnitřní orámování.



Ilustrace 3.1: Příloha D. vládního nařízení  
č. 203/1935 Sb. n.

Cena Kčs 1,20

Příloha k vyhlášce č. 141/1960 Sb.,  
kterou se vydávají pravidla silničního provozu

## SEZNAM DOPRAVNÍCH ZNAČEK

### I. DOPRAVNÍ ZNAČKY NA TABULÍCH (SVISLÉ)

#### A. VÝSTRAŽNÉ ZNAČKY



1 Příklad stružky nebo hrbit



2a Zatáčka vpravo



2b Zatáčka vlevo



2c Dvojitá zatáčka, první vpravo

1

Ilustrace 3.2: První strana vyhlášky  
141/1960 Sb. ze dne 3. září 1960



### 3.1 Trojúhelníkové dopravní značky

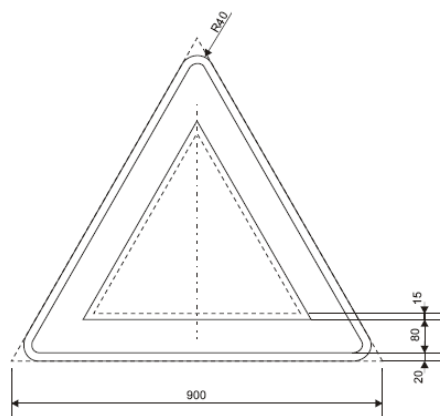
Podívejme se podrobněji na velikosti, typy a přesné specifikace trojúhelníkových značek.

Vzorové listy [7] předepisují následující velikosti:

- zmenšená – velikost strany trojúhelníku je 70cm
- základní – velikost strany trojúhelníku je 90cm
- zvětšená – velikost strany trojúhelníku je 125cm

Přesné rozměry pro jednotlivé části značek se mění s velikostí značky (obr.3.3).

Jistou komplikací může představovat charakter vzorových listů: „VL jsou zpracovávány ve formě otevřeného souboru, který je možno průběžně upravovat nebo doplňovat“ [7].



*Ilustrace 3.3: Náskres přesných  
proporcí trojúhelníkové značky  
základní velikosti*

Mezi trojúhelníkové značky patří 35 značek výstražných (skupina A) a 2 značky upravující přednost (skupina P). Dále je třeba uvažovat i značku O03 - Označení vozidla řízeného řidičem začátečníkem, která má velmi podobný vzhled, pročež ji musíme zařadit do procesu klasifikace, aby nedocházelo k jejímu zaměňování za značku jinou.

### 3.2 Rozdíly v dopravním značení

Jak již bylo zmíněno, dopravní značení není jednotné. Tento fakt znesnadňuje rozpoznávání dopravních značek a nutí nás vybrat si jednu ze dvou variant. První možností je navrhnout dostatečně robustní systém, který by bral v úvahu veškeré možné odlišnosti. Takový systém

by však byl velmi složitý, musel by být doplněn kvalitní snímací technikou a vyžadoval by velký výpočetní výkon. Pro ilustraci uvádím příklady zachycující rozdíly mezi značkami sousedních států (obr.3.4-3.7).



*Ilustrace 3.4: Ukázka německého dopravního značení*



*Ilustrace 3.5: Ukázka polského dopravního značení*

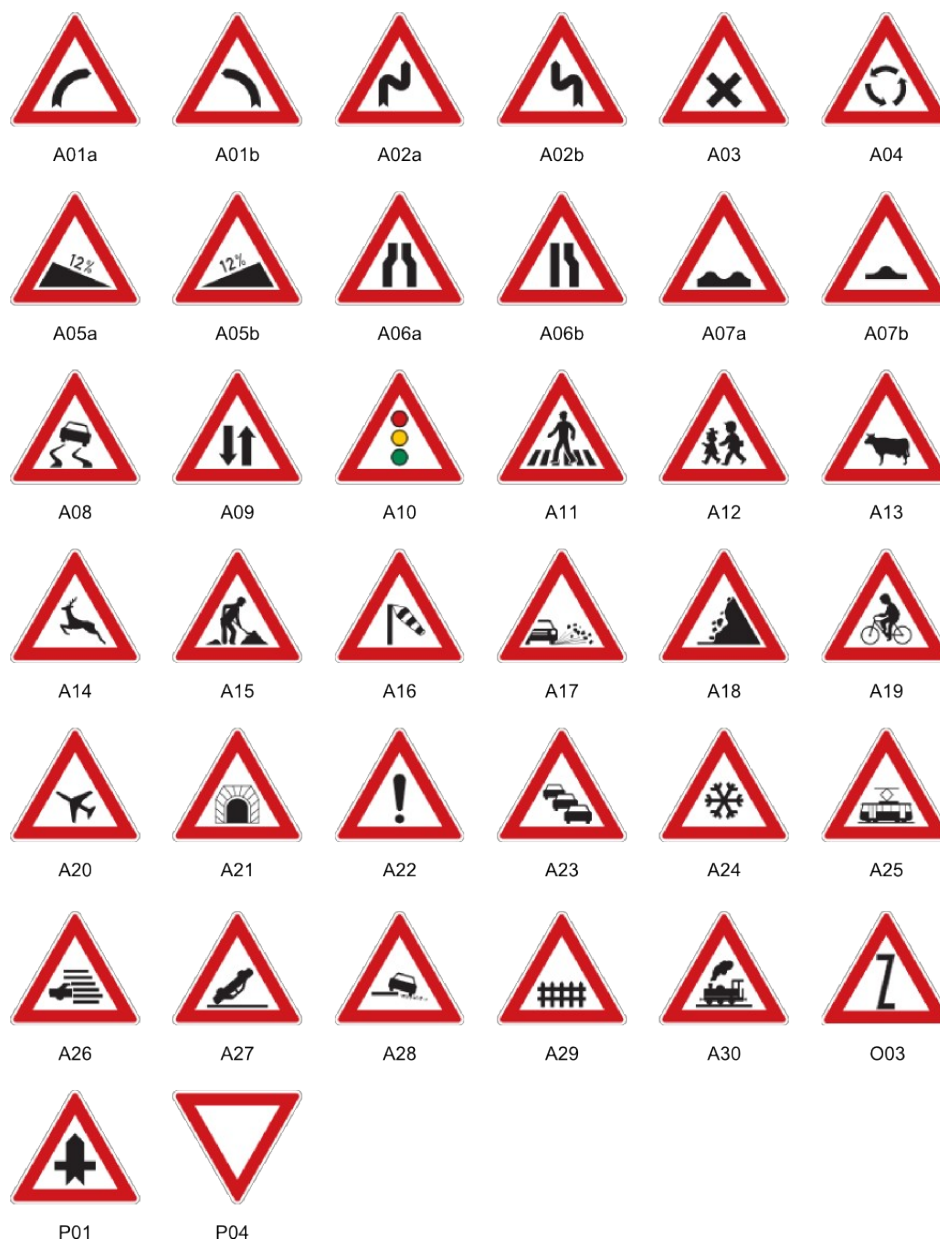


*Ilustrace 3.6: Ukázka rakouského dopravního značení*



*Ilustrace 3.7: Ukázka slovenského dopravního značení*

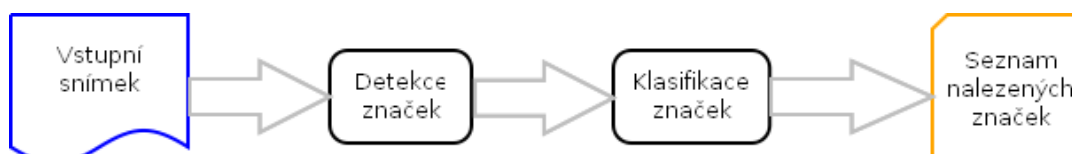
Druhá možnost spočívá v zaměření se na dopravní značení konkrétního státu (oblasti). Tato možnost byla zvolena i v této práci a rozpoznávané značky jsou aktuálně platným dopravním značením České republiky.



*Ilustrace 3.8: Všechny rozpoznávané značky*

## 4 Návrh systému

Základní dělení mého systému je, stejně jako v případě většiny podobných, na dvě základní části. První část zpracovává vstupní obraz ze zvoleného videosouboru a snaží se o detekci poloh dopravních značek. Druhou část systému tvoří klasifikátor, který určuje typ dopravních značek, které byly v obraze nalezeny.



*Ilustrace 4.1: Schema základního dělení*

### 4.1 Detekce značek

Pro vyhledávání značek jsem se rozhodl použít barevnou segmentaci a detekci hran. Barevná segmentace je výhodná především proto, že velká část plochy hledaných značek je červené barvy, která se ve zkoumaných snímcích jinak vyskytuje jen ojediněle. Segmentací omezíme plochu, na níž jsou značky hledány, na menší oblasti, které bývají označovány jako Region of Interest (ROI). Vyhledávání hran a trojúhelníkových tvarů v takto získaných oblastech již není výpočetně tak náročné, jako by to bylo při hledání v celém snímku. Stejně jako celý systém, tak i tato jeho část, je rozdělena na několik částí, kde každá má přesně specifikován svůj účel.

#### 4.1.1 Filtrace

Filtrace je potřeba ve dvou fázích detekce značek. Nejprve ji použijeme pro úpravu vstupních snímků, které mohou být rozostřené nebo jinak poškozené, ať už vlivem hardware, či vlivem prostředí. Použití jednoho či více filtrů by tak mělo sloužit k úpravě snímku, za účelem lepších výsledků při zpracování v dalších částech detektoru. Další fáze, při níž je potřeba užít filtrů, nastává po segmentaci. Segmentovaný obraz může obsahovat oblasti zájmu v podobě samostatných pixelů nebo jejich malých shluků. Naopak některé části obrazu mohou být po segmentaci rozděleny na více částí, což je také nutno korigovat. Za účelem filtrace je možno využít velkého množství existujících filtrů a jejich výběr je tak závislý především na kvalitě vstupních snímků a na tom, jaká data jsou vhodná pro další fáze detekce. Filtrační fázi by tedy mělo být možné jednoduše upravit a vhodnou kombinaci filtrů a jejich parametrů ověřit experimentálně.

### 4.1.2 Převod barevného prostoru

Před samotným hledáním červených oblastí v obraze je dobré převést obraz do jiného barevného schéma. Z klasického RGB tak snímek konvertujeme do HSV barevného prostoru. Výhodou HSV je především to, že barva je vyjádřena pouze první složkou, zatímco zbývající složky vyjadřují sytost a jas. Tento model se tak přibližuje tomu, jak barvy vnímá lidské oko. Z tohoto důvodu by HSV model měl být vhodnější právě proto, že dopravní značení je navrhováno s ohledem na přednosti a nedostatky lidského oka. Další výhoda HSV modelu pak tkví v tom, že nedochází k velké změně barevné složky se změnou světelných podmínek.

### 4.1.3 Segmentace

Jak již bylo zmíněno, segmentace určí plochy, na nichž se dále bude provádět vyhledávání značek. Pro tuto etapu je nejdůležitější najít vhodné hranice, které z množiny barev na snímcích vyčlení ty, které mohou být považovány za barvu dopravní značky. Po převodu na HSV je tento úkol zredukován na nalezení intervalu pro každou složku barevného modelu. Nalezení krajních mezi těchto intervalů je spíše experimentální úkol, který závisí na použitém snímacím zařízení a předpokládaných podmínkách, za kterých by měl tento systém operovat.

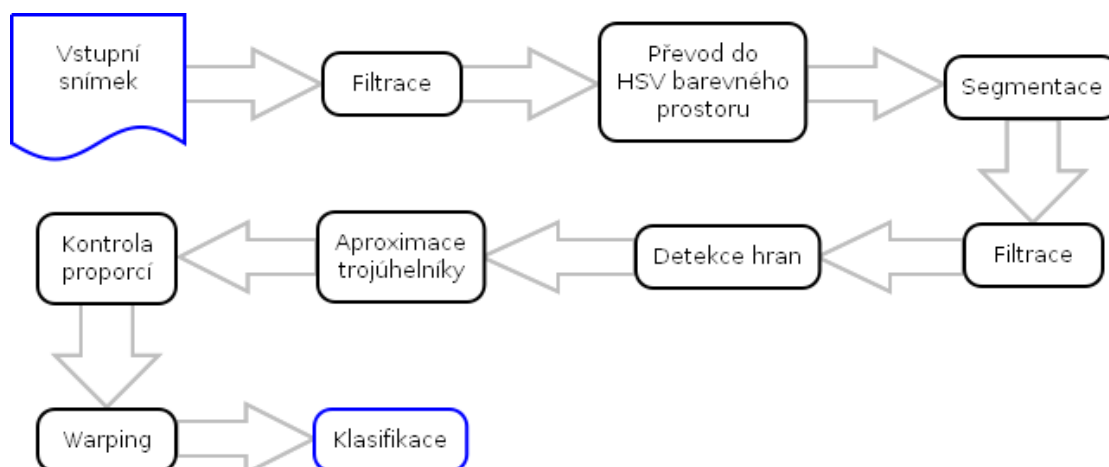
### 4.1.4 Detekce trojúhelníků

Výstupem předchozí a zároveň vstupem této fáze je bitová maska. Na tuto masku nejdříve použijeme některý z algoritmů pro detekci hran. Při výběru konkrétního algoritmu musíme brát ohled nejen na dosažený výsledek, ale i na dobu potřebnou pro jeho běh. Opět tak bude nejlepší implementovat algoritmů několik a jejich vhodnost se posoudí až experimentálně.

Po nalezení hran přichází na řadu aproximace těchto hran trojúhelníkem. Pokud se některé hrany podaří trojúhelníkem aproximovat, jedná se o trojúhelník. Vzhledem k tomu, že segmentační část algoritmu mohla vydat na výstupu i oblasti, které nejsou dopravními značkami, je třeba provést kontrolu. Tato kontrola by měla spočívat v analýze proporcí trojúhelníku, tj. například v porovnání délek jeho stran. Pokud při kontrole zjistíme, že jedna ze stran je výrazně kratší než zbylé dvě, nejedná se nejspíš o dopravní značku nebo je značka snímána z úhlu, který by mohl zapříčinit její chybné určení. Dále pak z dalšího zpracování můžeme vyloučit ty trojúhelníky, které neodpovídají svou velikostí. Příliš malá (vzdálená) značka by opět nemusela být správně určena klasifikátorem a snaha o její určení by tak jen zpomalovala celý algoritmus. Proto ji ze zpracování vyloučíme s tím, že se pravděpodobně objeví i na některém z dále zpracovávaných snímků ve větším měřítku. Vyloučit můžeme i dostatečně velké trojúhelníky, které značku představovat nemohou.

Nalezené trojúhelníky – potenciální značky – nakonec ještě musíme upravit tak, aby je bylo možné zpracovat klasifikátorem. Značka na snímku může být různě pootočená a zároveň

nemusí být v požadovaném měřítku, což napravíme metodou zvanou warping. Klasifikátoru je tak jako vstup poskytnut vždy stejně velký a stejně orientovaný obraz nalezené značky.



*Ilustrace 4.2: Schema průběhu detekce*

## 4.2 Klasifikace značek

K určení konkrétního typu značky jsem se rozhodl využít dvouvrstvé neuronové sítě. Tento typ neuronové sítě je jedním z nejčastěji užívaných, a to i při rozpoznávání obrazu. Jelikož neexistují jednoznačná kritéria pro výběr parametrů a velikosti takovéto sítě, držíme se pouze zavedených doporučení a příslušné hodnoty se upravují až na základě obdržených výsledků.

Do klasifikátoru budou vstupovat ty části snímku po segmentaci, které byly označeny za dopravní značky. Tyto části mají trojúhelníkový tvar a po uspořádání jednotlivých řádků pixelů do jedné souvislé posloupnosti slouží jako vstupní vektor neuronové sítě.

Proto, aby klasifikátor fungoval požadovaným způsobem, je nutno po jeho realizaci vytvořit trénovací množinu vstupů a k nim příslušnou, požadovanou množinu výstupů. Opakovaným předkládáním těchto množin neuronové síti dochází k jejímu učení. Během rozpoznávání obrazu, u tohoto typu neuronové sítě, již nedochází k dalšímu učení.

## 5 Detektor - použité algoritmy

### 5.1 Převod do HSV prostoru

Prvním krokem algoritmu je převod do výhodnější reprezentace obrazových dat. Stejně jako RGB tak i HSV barevný model reprezentuje barvy pomocí tří hodnot. První hodnotou je barevný tón, neboli odstín. Sytost představuje množství šedi v poměru k odstínu. Poslední hodnotou je jas udávající množství bílého světla v barvě. Jediná změna oproti RGB je jen ve velikosti intervalů.

	RGB		HSV	
	význam	interval	význam	interval
1. složka	Červený kanál (R)	0-255	Barevný tón (H)	0-360[°]
2. složka	Zelený kanál (G)	0-255	Sytost (S)	0-100[%]
3. složka	Modrý kanál (B)	0-255	Hodnota jasu (V)	0-100[%]

Tabulka 2: Intervaly hodnot barevných modelů

Jednotlivé složky je možno vypočítat následujícími vzorci pro nelineární transformaci[8]:

$$V = \max(R, G, B) \quad (2)$$

$$S = \begin{cases} (V - \min(R, G, B)) / V & \text{pokud } V > 0 \\ 0 & \text{pokud } V = 0 \end{cases} \quad (3)$$

$$H = \begin{cases} \vartheta & \text{pokud } \vartheta \geq 0 \\ 360 - \vartheta & \text{pokud } \vartheta < 0 \end{cases} \quad (4)$$

$$\vartheta = \begin{cases} (G - B) \cdot 60 / S & \text{pokud } V = R \\ (B - R) \cdot 60 / S + 180 & \text{pokud } V = G \\ (R - G) \cdot 60 / S + 240 & \text{pokud } V = B \end{cases} \quad (5)$$

Podle typu proměnné, do které vypočítané hodnoty ukládáme, a podle jejich následného využití si můžeme jejich rozsah upravit. Pokud se například ukládá do 8bitové proměnné, je

vhodné využít celý rozsah hodnot, které v ní mohou být uloženy. Pokud bychom rozsahy neupravily, nejenže by byla citelně zmenšena přesnost, ale v případě barevné složky (H) by došlo k podstatnému ořezání hodnoty.

## 5.2 Filtrace

Několik možných úprav snímku lze provést pomocí jediné funkce knihovny OpenCV a tou je CvSmooth. Tato funkce vyžaduje, jako jeden z povinných parametrů, typ požadované operace, který je dán některou z následujících konstant.

- CV\_BLUR – Lineární konvoluce
- CV\_GAUSSIAN – Konvoluce s Gaussovým jádrem
- CV\_MEDIAN – Konvoluce s výpočtem mediánu
- CV\_BILATERAL – Konvoluce zohledňující podobnost pixelů.

### 5.2.1 Konvoluce

Pro několik operací s obrazovými daty je možno použít matematickou operaci zvanou konvoluce. V algoritmu je využita před i po segmentaci obrazu a to k odstranění šumu a vyhlazení obrazu.

Konvoluce je důležitou operací, pro kterou bývá používán operátor \* a pro dvě funkce  $I(x)$  a  $h(x)$  ji můžeme definovat jako

$$I(x) * h(x) = \int_{-\infty}^{\infty} I(x - \alpha) h(\alpha) d\alpha \quad (6)$$

Kde funkce  $h(x)$  představuje jádro (masku) konvoluce. V oblasti digitálního zpracování obrazu se využívá diskrétní konvoluce, která má tuto podobu

$$I'_{i,j} = I_{i,j} * h_{i,j} = \sum_{x=-k}^k \sum_{y=-k}^k I_{i-x, j-y} h_{i,j} \quad (7)$$

Zde je možno chápat konvoluční jádro jako tabulku o rozměrech  $\langle -k, k \rangle \times \langle -k, k \rangle$ . Při aplikování této metody na obraz postupně prokládáme každý pixel konvolučním jádrem. Výsledná hodnota nového pixelu tedy vzniká jako vážený součet původního pixelu a jeho okolí. Jednotlivé váhy udává právě jádro konvoluce. To znamená, že počet vykonaných operací je



vždy závislý pouze na velikosti obrazu a jádra, zatímco obsah obrazu na něj nemá žádný vliv. Rychlost výpočtu pak bývá zvýšena hardwarovou akcelerací (např OpenGL 1.2), nebo paralelizací [9].

Metodě CvSmooth je třeba, pro provedení lineární konvoluce, předat dva parametry, které udávají velikost jádra. Váhy pro jednotlivé pixely jsou dány vztahem  $1/(\text{počet prvků jádra})$ , z čehož je patrné, že jde vlastně o průměr hodnot devíti pixelů. Efektem této operace je vyhlazení obrazu, které má však za následek rozmazání hran.

Gaussovo jádro konvoluce dostaneme, užijeme-li vztah pro Gaussovo neboli normální rozložení

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (8)$$

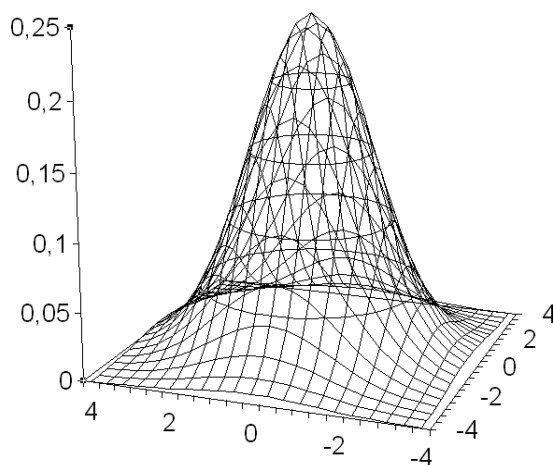
Parametr posunu  $\mu$  můžeme zanedbat a vzorec upravit pro dvourozměrný prostor

$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (9)$$

Pro velikost jádra  $3 \times 3$  dostáváme matici

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Konvoluce s Gaussovým jádrem se používá jako lineární nízkofrekvenční filtr pro odstranění šumu.



*Ilustrace 5.1: Normální rozdělení ve 2D*

Další typ konvoluce využívá mediánu, což je 50% kvantil. Pokud máme náhodnou veličinu  $x$ , pak jejím mediánem  $M$  je hodnota, pro kterou platí, že pravděpodobnost jevu  $x < M$  je jedna polovina. Jinými slovy je to hodnota, pro kterou platí, že 50% hodnot výběru je menší nebo je jí rovno. Při užití v konvoluci stačí hodnoty v lokálním okolí právě zpracovávaného bodu uspořádat podle velikosti a jako medián, a tedy i novou hodnotu tohoto obrazového bodu, zvolit prostřední hodnotu. Medián má tu výhodu oproti průměru, že je odolnější proti odlehlým hodnotám (pozorováním). Použití tohoto filtru však způsobuje porušení tenkých čar a ostrých rohů v obraze.

Poslední z použitých konvolučních filtrů se nazývá bilaterální a byl poprvé publikován v roce 1998 [10]. Opět se jedná o filtr vhodný k odstraňování šumu, na rozdíl od většiny ostatních filtrů však zachovává ostrost hran. Pokud je filtrována plocha s body obdobné barvy, chová se bilaterální filtr stejně jako gausův filtr. Je-li ale filtrována plocha v blízkosti hrany, pak se pro další výpočty použijí pouze pixely obdobné barvy, zatímco ostatní jsou ignorovány. Poznamenejme, že při filtrování barevného obrazu nestačí provést nezávisle filtrování každého ze tří kanálů. V takovém případě by došlo k zesílení nežádoucího míchání barev v barevných přechodech. Aby použití bilaterálního filtru bylo možné, musíme barevné obrazy nejprve převést do CIE-LAB barevného prostoru.

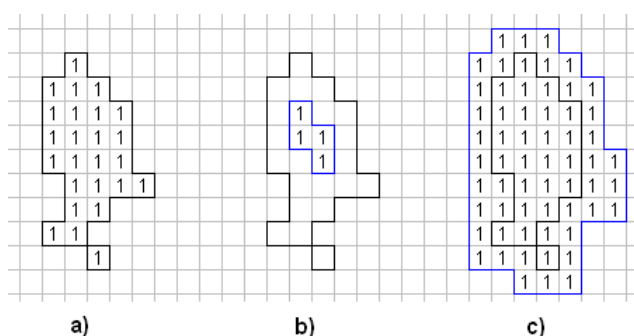


*Ilustrace 5.2: Ukázka bilaterálního filtrování, originál(vlevo), po filtraci(vpravo)*

### 5.2.2 Morfologická filtrace

Při zpracování snímku se může stát, že se vyskytnou velmi malé oblasti zájmu (ROI), které mohou být tvořeny jediným pixelem nebo malou skupinkou pixelů. Zpracováváním těchto oblastí bychom zbytečně ztráceli čas. K jejich nalezení a odstranění použijeme morfologický filtr.

Při morfologické filtraci se používá dvou operací a to eroze a dilatace. Pořadí operací je následující: eroze-dilatace-dilatace-eroze. Přestože obě dvě operace mohou být aplikovány na šedotónové obrázky, předvedeme si, pro zjednodušení, jejich princip na binárním obrazci. Algoritmus obou operací je podobný konvoluci a taktéž používá okénko o daném rozměru, které je postupně posouváno po celém obraze. V případě eroze bude nová hodnota pixelu rovna jedné pouze v případě, kdy byly nastaveny na tuto hodnotu i všechny pixely náležející do blízkého okolí daného velikostí okénka. V případě dilatace naopak nastavujeme pixel na hodnotu jedna, pokud je alespoň jeden z pixelů v blízkém okolí nastaven na hodnotu jedna. Popsané lze pozorovat na obrázku obr.5.3



*Ilustrace 5.3: Příklad eroze a dilatace*

### 5.3 Prahování

Prahování slouží k rozdělení pixelů na dvě množiny, kde první obsahuje pixely s hodnotou menší než prahová hodnota a do druhé množiny náleží všechny ostatní pixely. Obrazové body pak nabývají hodnoty podle příslušnosti k dané množině.

$$out_{i,j} = \begin{cases} \text{pokud } in_{i,j} > P & 1 \\ \text{jinak} & 0 \end{cases} \quad (10)$$

Použitím prahování získáme ke zpracovávanému obrazu bitovou masku, v níž nastavené pixely určují oblasti s požadovanou vlastností barvy.

### 5.4 Segmentace

Použitá metoda segmentace provádí prahování jednotlivých barevných kanálů v obraze. Pro každý kanál jsou určeny dvě prahové hodnoty a prahování tak probíhá na dvakrát. Jedna hodnota udává horní hranici a druhá dolní. Pokud při prahování používáme funkci, která přesně dodržuje vztah uvedený výše (např. v OpenCV), je nutno provést negaci výstupu prahování podle horní meze. Výsledné bitové mapy obou prahování je třeba sloučit, k čemuž stačí provést logický součin jednotlivých bitů (pixelů).

$$out = \begin{cases} \text{pokud } \begin{matrix} P_{H\_min} < in_H < P_{H\_max} \wedge \\ P_{S\_min} < in_S < P_{S\_max} \wedge \\ P_{V\_min} < in_V < P_{V\_max} \end{matrix} & 1 \\ \text{jinak} & 0 \end{cases} \quad (11)$$

Výsledná bitová mapa udává které barevné body mají požadované vlastnosti.

### 5.5 Hledání kontur

Při hledání obrysů objektů, které byly nalezeny segmentací obrazu, postupujeme následujícím způsobem.

1. Nejprve sekvenčně procházíme jednotlivé pixely obrazu a to tak dlouho, dokud nenajdeme první pixel objektu. Nalezený pixel nutně leží na hranici objektu.

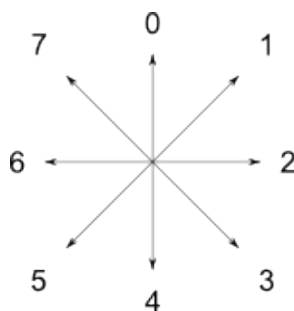
2. Sledujeme hranici objektu a to tak dlouho, dokud se nedostaneme do výchozího bodu, čímž se obrys uzavře. Sledování hranice funguje následovně.
  - Pokud se nacházíme uvnitř tělesa, směr pohybu po obrysu otočíme o  $90^\circ$  doleva a provedeme krok vpřed.
  - Nacházíme-li se vně tělesa, jehož hranice hledáme, otočíme se o  $90^\circ$  doprava a provedeme krok vpřed.

Pokud pro nalezení kontur použijeme metodu `cvFindContours` knihovny OpenCV, obdržíme seznam pixelů, které reprezentují okraj objektu. Vzhledem k sousednosti jednotlivých pixelů je pro reprezentaci výstupu této funkce využito Freemanovo kódování.

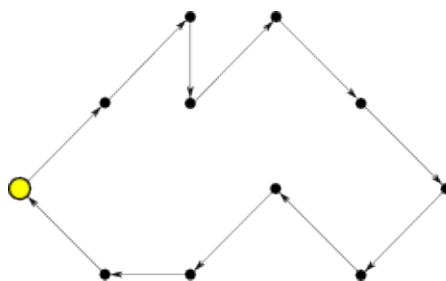
## 5.6 Freemanův kód

Aby nebylo nutné zaznamenávat souřadnice každého obrazového bodu tvořícího konturu, uložíme pouze souřadnice výchozího bodu a dále ukládáme pouze směr pohybu. Jako výchozí bod může sloužit první nalezený pixel objektu (viz. předchozí podkapitola). Pro uložení směru pohybu potřebujeme osm hodnot, což je dáno počtem sousedních pixelů.

Jako příklad slouží obrázek 5.4 znázorňující označení směrů a obrázek 5.5, který zachycuje příklad tvaru kontury. Výsledný řetězcový kód bude vypadat takto  $\{1, 1, 4, 1, 3, 3, 5, 7, 5, 6, 7\}$ .



*Ilustrace 5.4  
Označení směrů*

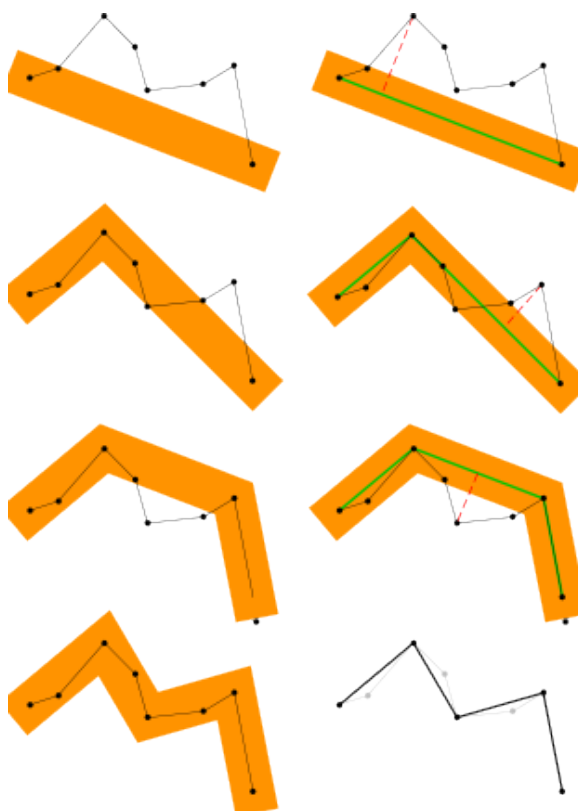


*Ilustrace 5.5 Příklad tvaru  
kontury*

## 5.7 Prokládání polynomem

Pro nalezení optimální aproximace nabízí OpenCV funkci `cvApproxPoly`, která využívá algoritmu Douglas-Peucker. Jde o rekurzivní využívající jako testovací oblast koridor o stanovené šířce. Průběh je následující:

1. Spoj první a poslední vrchol a vytvoř koridor.
2. Pokud všechny vrcholy neleží uvnitř koridoru, najdi vrchol, který je od koridoru nejvíce vzdálen. Pokud koridor obsahuje všechny vrcholy, algoritmus končí.
3. Bod nalezený v kroku 2 se stane vrcholem vznikajícího polygonu, čímž dojde k nahrazení jedné hrany dvěma novými. Na nově vzniklé hrany je rekurzivně aplikován krok 2.



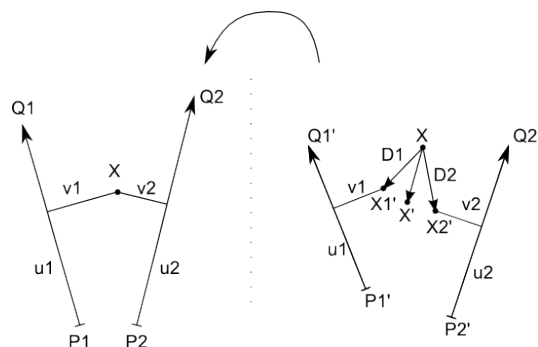
*Ilustrace 5.6 Ukázka aproximace*

## 5.8 Warping

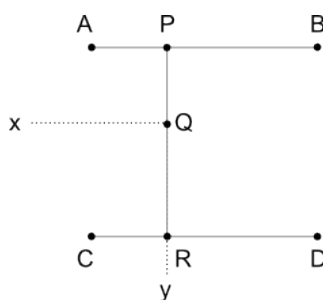
Warping je modifikací obrazu, která je prováděna na základě změny velikosti a polohy úseček. Úsečky se vyskytují v párech, kde jedna je pro vstupní obraz a druhá pro obraz výstupní. V průběhu algoritmu se pro každý pixel výstupního obrazu provádí následující:

1. Vypočítají se obrazy bodu podle jeho polohy vůči všem dvojicím úseček.
2. Pro každý obraz se vypočte vektor jeho posunutí vůči původnímu bodu.
3. Vypočtou se váhy jednotlivých obrazů.
4. Na základě obrazů a jejich vah se vypočte vážený součet, který je vektorem posunutí požadovaného a původního bodu.
5. Převzorkuje se oblast určená vypočtenými souřadnicemi.

První čtyři kroky uvedeného postupu nám vlastně řeknou, odkud z původního obrazu máme vzít informaci o intenzitě výstupního pixelu. Souřadnice, které jimi vypočteme, však nemusejí být celočíselné, proto provádíme krok č. 5. Převzorkování je realizováno bilineární interpolací, která počítá výslednou intenzitu pixelu jako vážený součet intenzit čtyř pixelů nejbližších vypočteným souřadnicím.



*Ilustrace 5.7 Ukázka warpingu podle dvou úseček*



*Ilustrace 5.8 Bilineární interpolace*

## 6 Klasifikátor – neuronová síť

### 6.1 Umělá neuronová síť

Umělé neuronové sítě se pokoušejí o napodobení struktury a funkce biologických neuronových sítí. Z důvodu velké složitosti těchto biologických předloh a velkého počtu neuronů, v nich obsažených, však zatím nejsme schopni simulovat je celé. Postupně se však daří simulovat alespoň jednodušší neuronové sítě, které mohou simulovat alespoň základní funkce např. lidského mozku.

Hlavní výhodou neuronových sítí je schopnost učit se. Učení probíhá na základě předkládání vzorů sítí, která nezná algoritmus řešení daného problému. Vývoj aplikací UNS tak přechází z hledání způsobu, jak danou úlohu algoritmizovat, na hledání trénovací množiny, kterou je možno síť naučit.

Další výhodou a důvodem použití právě NN ke klasifikaci dopravních značek je její schopnost generalizace. Pokud je síť správně naučena na korektní trénovací množině, dochází ke správnému rozpoznávání obrazu i pokud je na vstupu šum, či jiné zkreslení.

Na rozdíl od soudobých počítačů neuronové sítě, ať už umělé či biologické, tak striktně neoddělují zpracování informace od jejího uložení. Je možno říci, že zpracování, předávání a ukládání informací neprobíhá jen v jejích určitých částech, ale v celé síti [11]. Informace je v neuronové síti reprezentována silou vazby mezi neurony, která je při učení buď zesilována, nebo zeslabována.

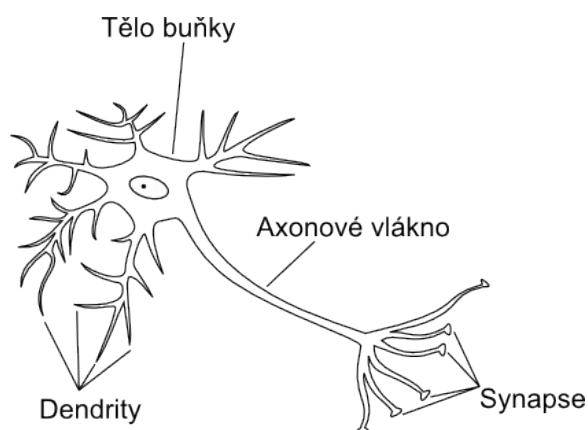
Umělé sítě se již dnes využívají pro řešení celé řady problémů v mnoha oborech. Uvedme alespoň několik příkladů jejich užití.

- Rozpoznávání a případná rekonstrukce obrazců. Kromě využití, kterému se věnuje tato práce, může se jednat například o rozpoznávání textu nebo otisků prstů.
- Modelování a řízení nelineárních systémů.
- Predikce časových řad v ekonomii, energetice, dopravě, lékařství, atd.
- Analýza signálů, jako je EKG a EEG.
- Kompresi, kódování a transformace signálů, především v telekomunikacích.



### 6.1.1 Neuron

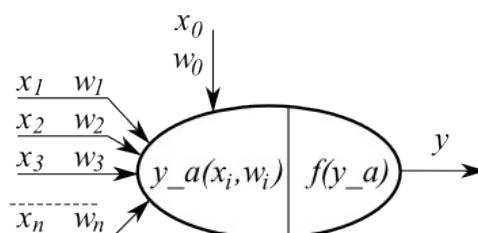
Neuron je základním stavebním prvkem všech neuronových sítí. Jedná se o specializovanou nervovou buňku, která se od ostatních biologických buněk liší především svým zvláštním tvarem. Z těla neuronu vystupují výběžky, které se nazývají dendrity a axiomy. Zatímco dendrity bývají krátké a rozvětvené, axiomy mohou dosahovat délky až 60cm[11].



*Ilustrace 6.1 Zjednodušené schema biologického neuronu*

V umělé neuronové síti se jeho funkce nahrazují funkcemi matematickými. Dodnes jedním z nejvýznamnějších a nejpoužívanějších modelů neuronů tzv. perceptron [12]. Model umělého neuronu na každém svém vstupu přijímá hodnotu, kterou násobí hodnotou příslušného prahu. Hodnota prahu může být od 0 (daný vstup neovlivňuje výstupní hodnotu neuronu) po 1 (daný vstup se na ovlivnění výstupu podílí v maximální možné míře). Poznamenejme, že vstup označovaný indexem 0, je využit jako práh neuronu a příslušná váha vždy nabývá hodnoty 1. Všechny vstupní hodnoty se po přenásobení sečtou, čímž dostáváme tzv. vstupní potenciál neuronu.

Výstup neuronu je dán aktivační funkcí, která nejčastěji mívá tvar gaussovské, hyperbolické, lineární, sigmoidální nebo skokové funkce. V literatuře, při popisu vztahů v NS, se nejčastěji setkáme se sigmoidální funkcí. Volba vhodné funkce záleží na typu konkrétní úlohy, případně na poloze neuronu v síti. Často bývá využito jiné aktivační funkce pro neurony ve skrytých vrstvách a jiné funkce pro neurony ve vrstvě výstupní[11]. Takovýchto buněk je v lidském mozku až 15 miliard a každý z nich může být napojen až na 5000 jiných neuronů.



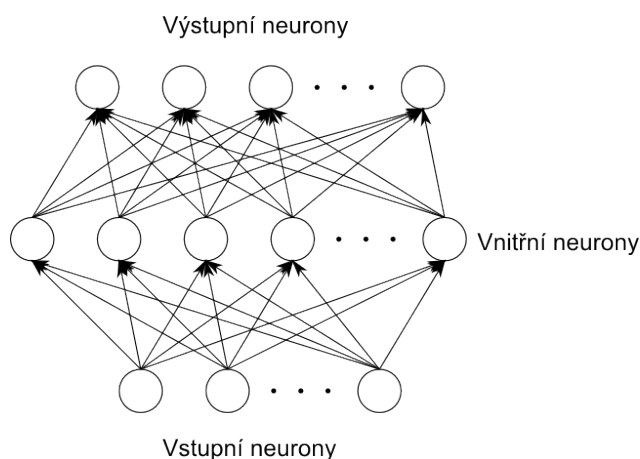
*Ilustrace 6.2 Schema umělého neuronu (perceptronu)*

### 6.1.2 Vícevrstvá neuronová síť

Vícevrstvé neuronové sítě bývají jedním z nejčastějších způsobů propojení umělých neuronů. V síti je jedna vrstva vstupních neuronů, několik skrytých vrstev a vrstva výstupní. Nejčastěji bývá užitá jedna, nebo dvě skryté vrstvy. Mezi každými dvěma sousedními vrstvami je úplné spojení neuronů, což znamená, že každý neuron vyšší vrstvy přijímá jako vstupy výstupy všech neuronů vrstvy nižší.

Odezva neuronové sítě se získává tzv. dopředným šířením (feedforward) signálu. Tento postup je popsán následujícími kroky[12]

1. Neurony vstupní vrstvy excitujeme na požadovanou úroveň danou vstupem sítě.
2. Výstupy těchto neuronů se přenesou na vstupy neuronů vyšší vrstvy, přičemž dojde k jejich upravení (zeslabení či zesílení) dle příslušných vah synapsí.
3. Každý neuron vyšší vrstvy vypočítá součet upravených vstupních signálů a na výstupu předá hodnotu danou jeho vnitřní (aktivační) funkcí.
4. Postup probíhá přes všechny vrstvy až k výstupní vrstvě. Odezva sítě je pak dána výstupy (hodnotami excitace) neuronů výstupní vrstvy.



*Ilustrace 6.3 Vícevrstvá neuronová síť*

### 6.1.3 Metoda učení - Backpropagation

Backpropagation, neboli metoda zpětného šíření, se používá k adaptaci sítě nad danou množinou trénovacích vzorů. Sítí procházíme v opačném směru než při dopředném šíření signálu a upravujeme hodnoty synaptických vah dle tohoto postupu[12]

1. Na vstup sítě umístíme vektor vstupních hodnot  $i$ -tého trénovacího vzoru.
2. Provede dopředné šíření signálu.
3. Srovnáme požadovanou odezvu sítě se skutečnými hodnotami výstupů, čímž obdržíme chybu sítě.
4. Chyba se v síti šíří směrem od výstupní k vstupní vrstvě a pomocí ní se mění hodnoty vah, čímž dochází ke zmenšování této chyby.
5. Po předložení všech trénovacích vzorů se vyhodnotí celková chyba sítě. Pokud je její hodnota větší než požadujeme, proces učení opakujeme.

Použitá neuronová síť slouží jako klasifikátor dopravních značek. Jejím vstupem je piktogram značky detekované v obraze a výstupem jsou hodnoty, které určují, nakolik daný vstup odpovídá jednotlivým značkám. Z toho vyplývá, že počet výstupů sítě/klasifikátoru je 38, což odpovídá počtu značek, které chceme rozpoznávat.

Metoda Backpropagation má několik variant, které se liší například způsobem výběru následujícího vzoru trénovací množiny, který se síť bude učit. Dále se může lišit to, jak často upravujeme hodnoty vah. Některé z metod při učení provádějí adaptaci parametrů, jako je koeficient učení, který určuje jak rychle se síť učí nové vzory.

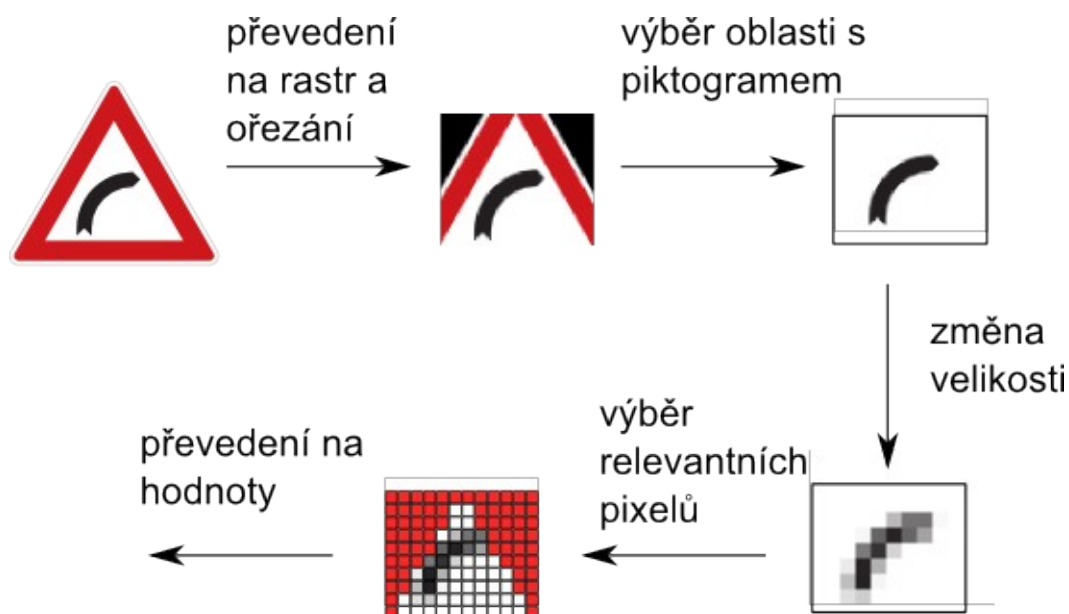
Při učení sítě se však ukázalo, že pro danou trénovací množinu je optimální standardní BP, která upravuje váhy po každém předložení vzoru.

## 6.2 Trénovací množina

Jako trénovací množina byly použity obrázky všech trojúhelníkových značek, které přesně odpovídají příslušné normě pro dopravní značení. Síť se tak učí vzory, které neobsahují šum a jiné případné nedostatky. Po naučení by síť měla umět, díky její schopnosti generalizace, rozpoznávat značky i ze snímků reálných značek.

Před samotným trénováním bylo nutno jednotlivé obrazové reprezentace značek transformovat do potřebné podoby. Jako první bylo vykonáno ořezání červeného okraje značky, který je vždy stejný a na rozhodnutí o typu značky nemá vliv. Zbývá část, která obsahuje piktogram, byla zmenšena na velikost 12x10 pixelů. Takováto velikost by měla být dostatečně malá, aby nebylo třeba příliš velká neuronové sítě, a zároveň by ještě měla umožnit určení

příslušnosti ke konkrétnímu typu dopravní značky. Jelikož použitá knihovna FANN vyžaduje jako vstup pro trénování sítě data v textové podobě, byly hodnoty pixelů převedeny ze stupňů šedi v intervalu 0-255 na reálná čísla z intervalu od -1 do 1. Vzhledem k trojúhelníkovému tvaru značky není potřeba brát v úvahu celou plochu obrázku, ale pouze tu část, která reprezentuje místo s piktogramy. Velikost této části byla, po prozkoumání velikosti a umístění piktogramů, stanovena na 54 pixelů z původních 120.



*Ilustrace 6.4 Ukázka tvorby trénovacího vzoru*

Vzhledem k možnosti užití různých filtrů na obraz a z toho plynoucích důsledků, jako je především posun nalezených hran ve značce, byly do trénovací množiny zařazeny i vzory, kde jsou piktogramy posunuty o jeden pixel do všech směrů od své původní polohy. Tím došlo k nárůstu trénovací množiny z 38 na 342 vzorů.

Další nárůst trénovací množiny byl zapříčiněn přidáním dalších vzorů, které byly navíc obohaceny o náhodný šum. Šum byl přidáván proto, že se při testování projevovala velice negativně přítomnost šumu na vstupu sítě a nedocházelo ke správné klasifikaci. Přidání šumu by mělo vést k lepšímu nastavení hodnot vah v síti a k lepší klasifikaci při testování.

### 6.3 Trénování klasifikátoru

Ve fázi učení bylo kromě samotného naučení sítě potřeba stanovit několik dalších parametrů sítě. Nejdůležitějším z nich byl počet skrytých vrstev neuronové sítě a počet neuronů v těchto vrstvách obsažený. Jak ukazují zkušenosti, většina modelovaných soustav bývá dostatečně

správně aproximována již při použití jediné skryté vrstvy [11]. Pokud by bylo v síti příliš mnoho skrytých vrstev, nebo by tyto vrstvy obsahovaly příliš mnoho neuronů, nemusí to nutně znamenat zkvalitnění výsledků, které obdržíme. V takovýchto případech naopak dochází k tzv. „přeučení sítě“, které se projevuje tak, že natrénovaná síť rozpoznává správně jen vzory trénovací množiny, ale při jejich drobné změně je nevyhodnotí korektně.

Dalšími parametry, které lze při použití knihovny FANN navolit, jsou učicí metoda a aktivační funkce neuronů. Pro optimální určení těchto parametrů byla použita utilita FannTool, která provádí testování vlivu nastavení možných kombinací parametrů na učení sítě.

Po vyhodnocení všech testů a nastavení optimálních parametrů, přichází na řadu samotné učení sítě, které trvá řádově několik set epoch, než obdržíme požadovanou celkovou odezvu sítě na trénovací množinu.

## 7 Testování výsledné aplikace

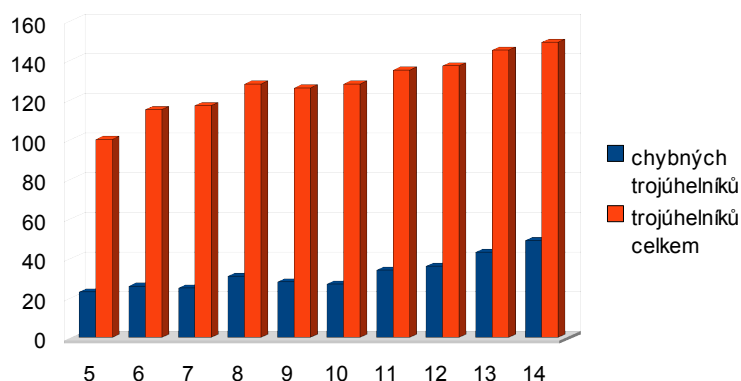
### 7.1 Testovací data

Pro testování byla použita data ze systému na jehož vývoji se VŠB podílí, která byla tvořena videosoubory a snímky pořízenými na komunikacích v okolí Ostravy-Poruby a německého Wolfsbugru. Z původních asi 11000 snímků bylo vybráno 600 snímků, na kterých je zachycena nějaká trojúhelníková dopravní značka. Následně byly snímky uloženy jako videosekvence, ke které byl vytvořen textový soubor s výčtem značek na snímcích. Tento textový soubor slouží k porovnání výstupu neuronové sítě s hodnotami, kterých chceme dosáhnout.

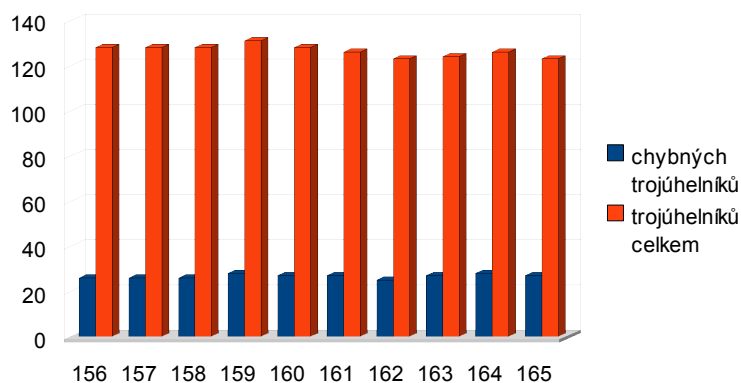
Textový soubor s popisem značek v testovacím souboru obsahuje název značky i tehdy, je-li značka malá, či nečitelná. Již při vytváření textového souboru se proto počítalo se spíše menší úspěšností, ve smyslu nalezení všech „označených“ dopravních značek.

### 7.2 Intervaly hodnot pro segmentaci

Jak již bylo napsáno dříve, pro segmentaci potřebujeme určit tři intervaly hodnot, které budou reprezentovat červenou barvu značek. Interval pro složku intenzity a sytosti barvy nám umožňují ze snímku vyloučit příliš tmavé a příliš světlé barvy. Jejich intervaly by měly být co možná největší, abychom vyloučili pouze extrémní hodnoty. Zvolení konkrétních hodnot závisí na světelných podmínkách scény při pořizování snímku a je spíše experimentální. Mnohem zajímavější, a na určení náročnější, je interval pro barevný tón. Velikost hodnot byla nejprve určena experimentálně na základě převodu několika snímků do HSV barevného modelu a poté byly porovnávány výsledky detekce při změně velikosti intervalu. Vliv změny hodnoty prahů je zachycen na následujících grafech.



*Graf vlivu hodnoty dolní meze na počet detekovaných trojúhelníků*



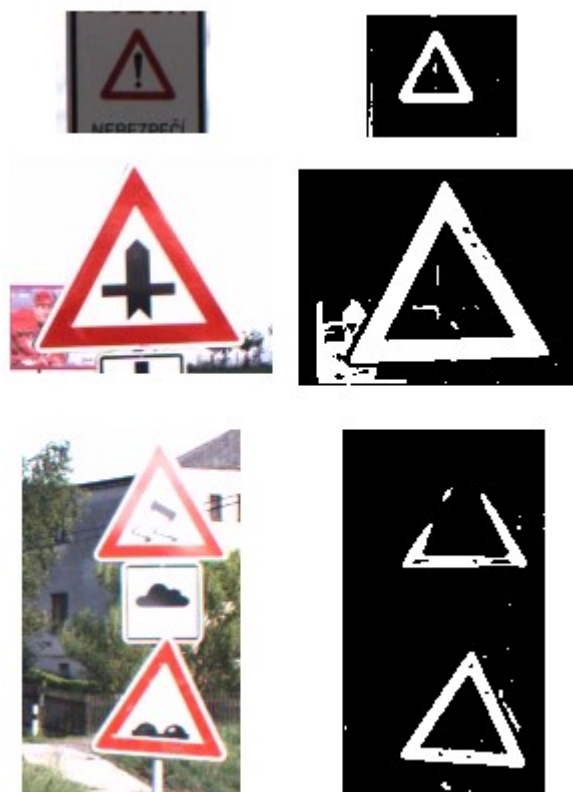
*Graf vlivu hodnoty horní meze na počet detekovaných trojúhelníků*

Zatímco při pohybu horní meze nedocházelo ke znatelnější změně počtu detekovaných trojúhelníků, při zvětšování spodního prahu se od hodnoty 10 začal zvyšovat jak celkový počet nalezených trojúhelníků, tak počet trojúhelníků, které rozměrově neodpovídají značce.

	prahy	
	dolní	horní
Barevný tón (H)	10	160
Sytost (S)	63	235
Hodnota jasu (V)	3	254

*Tabulka 3: Hodnoty prahů pro segmentaci*

Použití uvedených hodnot prahů vede k poměrně dobrým výsledkům pokud je značka tmavší, ale problémy nastávají v případech, kdy je značka silněji osvětlena. Obrázek zachycuje několik příkladů



*Ilustrace 7.1 příklady výsledků segmentace*

### 7.3 Výběr filtrů

Jako poměrně úspěšná se ukázala kombinace gaussova filtru ve fázi před segmentací obrazu, spolu s morfologickým filtrem a jemným rozmazáním po segmentaci. Jelikož však nebyly náležitě otestovány všechny možné kombinace filtrů, z důvodu enormního množství možností, nemusí jít nutně o tu nejlepší. Na následujících obrázcích jsou zachyceny fáze průběhu algoritmu, při nichž se filtrace uplatňuje.





*Ilustrace 7.2 Zdrojový obraz*



*Ilustrace 7.3 Výstup segmentace*



*Ilustrace 7.4 Obraz po odstranění šumu*

## 7.4 Úspěšnost detekce značek

Během testování algoritmu na sekvenci snímků byla úspěšnost malá a k nalezení trojúhelníku došlo pouze na 46 snímcích z celkových 583. Zde by se mohlo zdát, že je to velmi málo, je si však třeba uvědomit, že snímky byly pořizovány z různých vzdáleností a na většině snímků je značka velmi malá.

Naproti tomu při analyzování videosekvence, ve které se značky postupně přibližovaly, bylo nejméně jedenkrát rozpoznáno 28 dopravních značek a pouze 6 nebylo odhaleno vůbec. Nedetekování některých značek bylo způsobeno světelnými podmínkami ve snímku. Příklad málo a hodně osvětlené značky zachycuje následující obrázek.



*Ilustrace 7.5 Nedetekované značky*

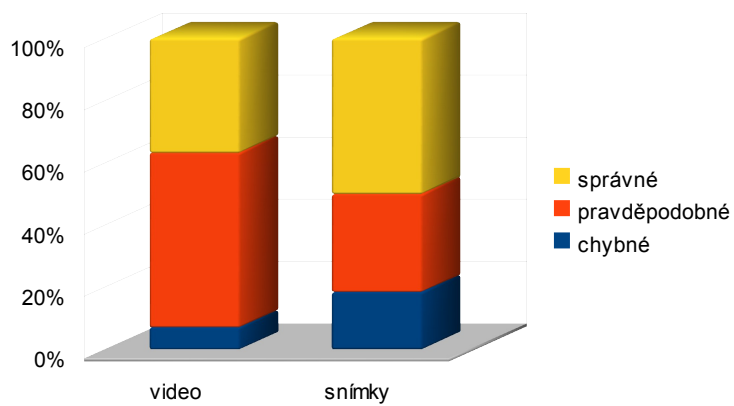
Několikrát došlo k nalezení trojúhelníku i v oblasti, kde se žádná značka nenacházela. Všechny takové trojúhelníky však byly z dalšího zpracování vyloučeny, z důvodu nesprávných rozměrů.



*Ilustrace 7.6 Příklad rozměrově nevyhovujícího trojúhelníku*

## 7.5 Úspěšnost klasifikace značek

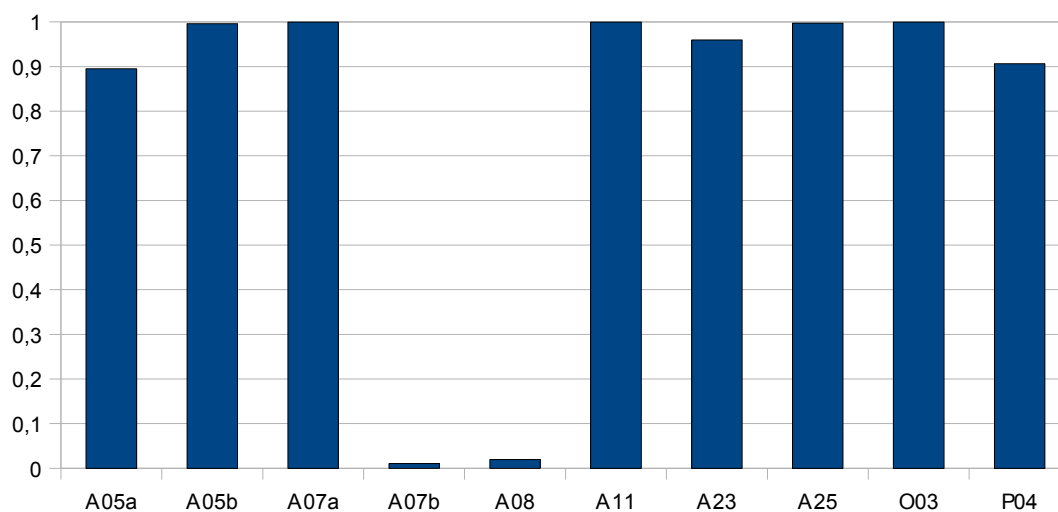
Klasifikace značek pomocí popsaného postupu se ukázala jako ne příliš úspěšná. Přesto, že učení sítě probíhalo bez problémů, ne vždy byly značky identifikovány jednoznačně. Ve většině případů se však skutečný typ značky objevil mezi typy, které byly neuronovou sítí určeny za nejpravděpodobnější. Relativní četnost správnosti určení značky zachycuje následující graf.



*Graf relativní úspěšnosti klasifikace značek*



*Ilustrace 7.7 Příklad úspěšné identifikace značek*



*Graf desítky nejpravděpodobnějších typů při klasifikaci značky A07a*

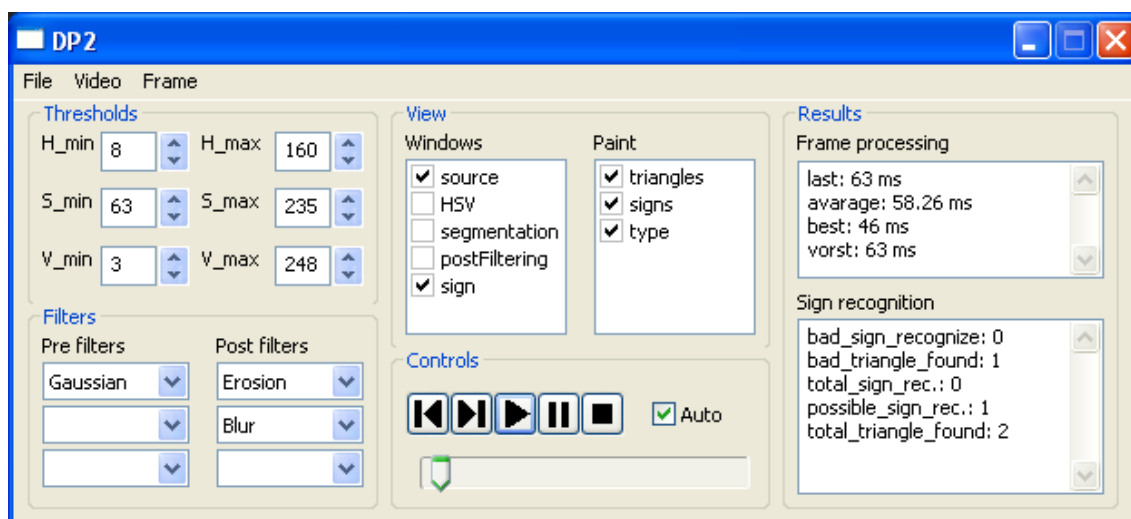
## 7.6 Rychlost aplikace

Během testování se ukázalo, že čas potřebný pro zpracování snímku se pohybuje okolo 50ms, což je však závislé na hodnotách parametrů, výběru použitých filtrů a samozřejmě na výkonu testovacího počítače. Při spuštění automatického testování tak video bylo zpracováváno rychlostí srovnatelnou s rychlostí přehrávání.

Při pokusu o měření rychlosti klasifikace bylo zjištěno, že NS předkládá výsledky za dobu kratší, než 1ms. Takto krátký časový úsek je poněkud obtížněji měřitelný a zároveň je, vzhledem k celkové době zpracování snímku, zanedbatelný.

## 8 Ovládání aplikace

Po spuštění se objeví hlavní okno aplikace, které je členěno do několika boxů se specifickým účelem.



*Ilustrace 8.1: Ukázka aplikace*

- **Thresholds** – Umožňuje nastavit hodnoty prahů pro segmentaci. Je třeba si uvědomit, že interval hodnot pro první barevnou složku je invertován. To znamená, že čím bližší si hodnoty prahů jsou, tím větší je interval. Nastaveny mohou být libovolné hodnoty z intervalu 0-255.
- **Filters** – V prvním sloupci můžeme nastavit až tři filtry použité na obraz před segmentací a v pravém sloupci filtry použité po segmentaci. Výběr filtrů není nijak omezen a výběr jejich správné(funkční) kombinace je zcela v rukou uživatele.
- **View** – Pomocí tohoto panelu si můžeme zobrazit okna s jednotlivými fázemi algoritmu. Pokud je zobrazeno okno se zdrojovým videem, můžeme si v něm nechat vykreslovat nalezené trojúhelníky a značky. Každý nalezený trojúhelník je orámován zelenou barvou. Pokud má trojúhelník navíc správný rozměr, označí se jeho rohy červenými kroužky. Po rozeznání značky je vypsán nejpravděpodobnější typ.
- **Controls** – Slouží k ovládání videa, tak jak je to běžné ve většině přehrávačů. Zaškrtnutí políčka auto způsobí při přehrávání automatické provádění detekce značek.
- **Results** – V horní části můžeme vidět časy a v dolní výsledky zpracování snímků. Časy, které jsou měřeny představují čas zpracování posledního snímku, průměrný čas, nejlepší a nejhorší čas. Mezi výsledky jsou tyto hodnoty:

- Chybně určené značky (bad\_sign\_recognized)
- Trojúhelníky neodpovídající rozměrům značky (bad\_triangle\_found)
- Počet nalezených a správně určených značek (total\_sign\_rec.)
- Nejednoznačně určené značky (possible\_sign\_rec.)
- Celkový počet nalezených trojúhelníků (total\_triangle\_found)

Pomocí menu aplikace je možno otevřít dialog pro výběr videa, nebo obrázku pro analýzu. Pokud je otevíráno video, aplikace se pokusí nalézt ve stejném adresáři i textový soubor se seznamem značek. Pokud soubor není nalezen, musí být analýza (testování) provedena ručně.

## 9 Závěr

Jako největší výhodu svého algoritmu vidím především jeho rychlost, která umožňuje zpracování všech snímků videosouboru, aniž by se musely některé snímky přeskakovat, a to i v reálném čase. Tímto tedy zůstává zachována možnost úpravy, nebo nahrazení některých algoritmů za robustnější, od čehož se dá očekávat lepší výsledek při odpovídajícím zpomalení rychlosti.

Ve fázi detekce značek by snad bylo možno označit výsledky za uspokojivé, nicméně použitá technika detekce má značné problémy s poškozenými značkami a při i částečném překrytí, značka není nalezena vůbec. Použití prahování při segmentaci také není úplně ideální, především pro fixní nastavení hodnot prahů. Jako možné vylepšení bych viděl využití algoritmu, který by se adaptoval podle intenzity osvětlení snímku.

Neuronová síť, ať už byla pro trénování použita jakákoli z trénovacích množin, neprojevila příliš svou schopnost generalizace. Trénování sice probíhalo po určení a nastavení příslušných parametrů poměrně rychle a přesně, zato při fázi testování jen málokdy došlo k získání stoprocentního výsledku. Výstupem sítě většinou bylo několik pravděpodobných typů značky, mezi nimiž se jen výjimečně nenacházel typ námi požadovaný. Nejpravděpodobnějšími příčinami tohoto jevu mohla být přítomnost šumu nebo jisté zkreslení piktogramu vzniklé při transformaci dat. K nápravě tohoto problému by snad mohlo přispět další rozšíření trénovací množiny. Jako další vzory by mohly sloužit piktogramy značek, které by se opět doplnily o nějaký typ šumu nebo zkreslení.

V případě, že by právě popsané navýšení počtu vzorů trénovací množiny nepřineslo dostatečné zlepšení, bylo by nutno použít jiný způsob klasifikace. Jinou rozšířenou možností klasifikace obrazu je užití tzv. příznaků, což jsou charakteristiky, které nabývají různých hodnot pro různé třídy vzorů. Po ustanovení vhodné množiny příznaků se nad ní stanovují hodnoty jednotlivých charakteristik a ty se posílají do neuronové sítě.

Nejenom rozpoznávání dopravních značek, ale i celý obor analýzy obrazu, je velice zajímavý a zajisté také perspektivní. Není tomu ani divu, když dopravní a jiná značení bývají převážně vizuálního charakteru a neustále roste potřeba jejich počítačového zpracování. Každý takovýto systém může znamenat zvýšení bezpečnosti a omezit tak případné škody zapříčiněné nepozorností řidiče.

Asi největším nepřítelem při rozpoznávání dopravního značení však zatím stále zůstává jeho nejednotnost. Vítanou pomocí by tak bylo postupné sjednocení vzhledu dopravních značek v jednotlivých státech.

Pokud by se podařilo úspěšně vyvinout algoritmus pro rozpoznávání veškerého dopravního značení, znamenalo by to velký skok kupředu a mohlo by to vést až k dopravním prostředkům řízeným pouze počítačem, bez nutnosti zásahu člověka – řidiče. Takovéto algoritmy však

i nadále zůstávají hudbou budoucnosti a k jejich vývoji ještě bude nutno vynaložit mnoho úsilí a prostředků.



## Seznam ilustrací:

Ilustrace 3.1: Příloha D. vládního nařízení č. 203/1935 Sb. n.....	9
Ilustrace 3.2: První strana vyhlášky 141/1960 Sb. ze dne 3. září 1960.....	9
Ilustrace 3.3: Nákres přesných proporcí trojúhelníkové značky základní velikosti.....	10
Ilustrace 3.4: Ukázka německého dopravního značení.....	11
Ilustrace 3.5: Ukázka polského dopravního značení.....	11
Ilustrace 3.6: Ukázka rakouského dopravního značení.....	11
Ilustrace 3.7: Ukázka slovenského dopravního značení.....	11
Ilustrace 3.8: Všechny rozpoznávané značky.....	12
Ilustrace 4.1: Schema základního dělení.....	13
Ilustrace 4.2: Schema průběhu detekce.....	15
Ilustrace 5.1: Normální rozdělení ve 2D.....	19
Ilustrace 5.2: Ukázka bilaterálního filtrování, originál(vlevo), po filtraci(vpravo).....	20
Ilustrace 5.3: Příklad eroze a dilatace.....	20
Ilustrace 5.4 Označení směrů.....	22
Ilustrace 5.5 Příklad tvaru kontury.....	22
Ilustrace 5.6 Ukázka aproximace.....	23
Ilustrace 5.7 Ukázka warpingu podle dvou úseček.....	24
Ilustrace 5.8 Bilineární interpolace.....	24
Ilustrace 6.1 Zjednodušené schema biologického neuronu.....	26
Ilustrace 6.2 Schema umělého neuronu (perceptronu).....	27
Ilustrace 6.3 Vícevrstvá neuronová síť.....	27
Ilustrace 6.4 Ukázka tvorby trénovacího vzoru.....	29
Ilustrace 7.1 příklady výsledků segmentace.....	33
Ilustrace 7.2 Zdrojový obraz.....	34
Ilustrace 7.3 Výstup segmentace.....	34
Ilustrace 7.4 Obraz po odstranění šumu.....	34
Ilustrace 7.5 Nedetekované značky.....	35
Ilustrace 7.6 Příklad rozměrově nevyhovujícího trojúhelníku.....	35
Ilustrace 7.7 Příklady úspěšné identifikace značek.....	36
Ilustrace 8.1: Ukázka aplikace.....	38

## Seznam tabulek

Tabulka 1: Intervaly barevných složek.....	6
Tabulka 2: Intervaly hodnot barevných modelů.....	16
Tabulka 3: Hodnoty prahů pro segmentaci.....	32

## Seznam použité literatury

1. PACLÍK, Pavel. The Automatic Classification of Road Signs. [s.l.], 1998. 74 s. Diplomová práce. České vysoké učení technické, Fakulta dopravní. Dostupné z WWW: <<http://euler.fd.cvut.cz/publikace/files/thesis.pdf>>.
2. LUTONSKÝ, Marek. Blaupunkt TravelPilot : navigace s živou kamerou. Navigovat.cz [online]. 14.10.2008, [cit. 2010-05-03]. Dostupný z WWW: <<http://navigovat.mobilmania.cz/Clanky/AR.asp?ARI=114048>>.
3. HATZIDIMOS, John. AUTOMATIC TRAFFIC SIGN RECOGNITION IN DIGITAL IMAGES. [s.l.], 2004. Diplomová práce. Aristotle University of Thessaloniki.
4. JOHANSSON, Bjorn. Road Sign Recognition from a Moving Vehicle. [s.l.], 2002. 72 s. Diplomová práce. Uppsala University. Dostupné z WWW: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.2984&rep=rep1&type=pdf>>.
5. KHAN, M. Taha. Real-Time Recognition System for Traffic Signs. [s.l.], 2008. 79 s. Diplomová práce. Dalarna University. Dostupné z WWW: <<http://dalea.du.se/theses/archive/7c46e67a-c368-470e-9430-447dfe6b1654/a72074eb-f0c9-4602-b2ae-c4a7c58db05d.pdf>>.
6. KUBÁLKOVÁ, Petra. Dopravní značky v Evropě. FiftyFifty.cz : společenský magazín [online]. 2008, 999, [cit. 2010-05-03]. Dostupný z WWW: <<http://www.fiftyfifty.cz/Dopravni-znacky-v-Evropě-1818874.php>>.
7. VZOROVÉ LISTY STAVEB POZEMNÍCH KOMUNIKACÍ : SVISLÉ DOPRAVNÍ ZNAČKY. [s.l.] : MINISTERSTVO DOPRAVY A SPOJŮ ČR ODBOR POZEMNÍCH KOMUNIKACÍ, 9.7.2001. 9 s. Dostupné z WWW: <<http://old.cdv.cz/text/projekty/dopznac/vl6-1-01.pdf>>.
8. OpenCV Reference Manual [online]. v2.1. [s.l.] : [s.n.], 2010 [cit. 2010-05-03]. Dostupné z WWW: <<http://ce.sharif.edu/courses/85-86/2/ce712/resources/root/OpenCV/OpenCV%20Tutorial.pdf>>.
9. ŽÁRA, Jiří; BENEŠ, Bedřich; SOCHR, Jiří. Moderní počítačová grafika. 2., přeprac. a rozš. vyd. Praha : Computer Press, 2004. 609 s. ISBN 80-251-0454-0.
10. TOMASI, Carlo; MANDUCHI, Roberto. Bilateral Filtering for Gray and Color Images [online]. [s.l.] : [s.n.], 1998 [cit. 2010-05-03]. Dostupné z WWW: <<http://www.cs.duke.edu/~tomasi/papers/tomasi/tomasiIccv98.pdf>>.
11. TAUFER, I.; DRÁBEK, O.; SEIDL, P. Umělé neuronové sítě : základy teorie a aplikace. CHEMagazín. 2006, XVI, č. 5, s. 12-14. Dostupný také z WWW: <[http://www.chemagazin.cz/Texty/CHXVI\\_5\\_cl6.pdf](http://www.chemagazin.cz/Texty/CHXVI_5_cl6.pdf)>.
12. VONDRÁK, Ivo. Umělá inteligence a neuronové sítě. [s.l.], 2009. 139 s. Skripta. Vysoká škola báňská-technická, Univerzita Ostrava.

## **Příloha: CD**

CD příloha obsahuje:

- **Elektronickou verzi textu této diplomové práce**

Text je uložen ve formátu PDF (Adobe® Acrobat Reader) a ODT (OpenOffice.org 3.0.0).

- **Vytvořenou aplikaci**

Obsahuje spustitelnou aplikaci, zdrojové kódy.

- **Trénovací množinu dat**

Kolekci obrázků znázorňující postupnou transformaci obrázků dopravních značek na trénovací data pro NN a aplikaci pro tvorbu vzorů.

- **Testovací data**

Představuje část dat, na kterých byla síť testována. Přiložena je i aplikace usnadňující tvorbu textových souborů s popisem testovacích videí.